Jim Lambers
MAT 610
Summer Session 2009-10
Lecture 4 Notes

These notes correspond to Sections 3.1 and 3.2 in the text.

Gaussian Elimination and Back Substitution

The basic idea behind methods for solving a system of linear equations is to reduce them to linear equations involving a single unknown, because such equations are trivial to solve. Such a reduction is achieved by manipulating the equations in the system in such a way that the solution does not change, but unknowns are eliminated from selected equations until, finally, we obtain an equation involving only a single unknown. These manipulations are called *elementary row operations*, and they are defined as follows:

- Multiplying both sides of an equation by a scalar
- Reordering the equations by interchanging both sides of the ith and jth equation in the system
- Replacing equation i by the sum of equation i and a multiple of both sides of equation j

The third operation is by far the most useful. We will now demonstrate how it can be used to reduce a system of equations to a form in which it can easily be solved.

Example Consider the system of linear equations

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 5, \\ 3x_1 + 2x_2 + 4x_3 &= 17, \\ 4x_1 + 4x_2 + 3x_3 &= 26. \end{aligned}$$

First, we eliminate x_1 from the second equation by subtracting 3 times the first equation from the second. This yields the equivalent system

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 5, \\ -4x_2 + x_3 &= 2, \\ 4x_1 + 4x_2 + 3x_3 &= 26. \end{aligned}$$

Next, we subtract 4 times the first equation from the third, to eliminate x_1 from the third equation as well:

$$x_2 + 2x_2 + x_3 = 5,$$

$$\begin{aligned} -4x_2 + x_3 &= 2, \\ -4x_2 - x_3 &= 6. \end{aligned}$$

Then, we eliminate x_2 from the third equation by subtracting the second equation from it, which yields the system

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 5, \\ -4x_2 + x_3 &= 2, \\ -2x_3 &= 4. \end{aligned}$$

This system is in *upper-triangular form*, because the third equation depends only on x_3, and the second equation depends on x_2 and x_3.

Because the third equation is a linear equation in x_3, it can easily be solved to obtain $x_3 = -2$. Then, we can substitute this value into the second equation, which yields $-4x_2 = 4$. This equation only depends on x_2, so we can easily solve it to obtain $x_2 = -1$. Finally, we substitute the values of x_2 and x_3 into the first equation to obtain $x_1 = 9$. This process of computing the unknowns from a system that is in upper-triangular form is called *back substitution.* □

In general, a system of n linear equations in n unknowns is in upper-triangular form if the ith equation depends only on the unknowns $x_i, x_{i+1}, \ldots, x_n$, for $i = 1, 2, \ldots, n$.

Now, performing row operations on the system $A\mathbf{x} = \mathbf{b}$ can be accomplished by performing them on the *augmented matrix*

$$\left[\begin{array}{cc} A & \mathbf{b} \end{array}\right] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right].$$

By working with the augmented matrix instead of the original system, there is no need to continually rewrite the unknowns or arithmetic operators. Once the augmented matrix is reduced to upper triangular form, the corresponding system of linear equations can be solved by back substitution, as before.

The process of eliminating variables from the equations, or, equivalently, zeroing entries of the corresponding matrix, in order to reduce the system to upper-triangular form is called *Gaussian elimination.* The algorithm is as follows:

for $j = 1, 2, \ldots, n-1$ **do**
  for $i = j+1, j+2, \ldots, n$ **do**
    $m_{ij} = a_{ij}/a_{jj}$
    for $k = j+1, j+2, \ldots, n$ **do**
      $a_{ik} = a_{ik} - m_{ij}a_{jk}$

```
        end
        b_i = b_i - m_ij b_j
    end
end
```

This algorithm requires approximately $\frac{2}{3}n^3$ arithmetic operations, so it can be quite expensive if n is large. Later, we will discuss alternative approaches that are more efficient for certain kinds of systems, but Gaussian elimination remains the most generally applicable method of solving systems of linear equations.

The number m_{ij} is called a *multiplier*. It is the number by which row j is multiplied before adding it to row i, in order to eliminate the unknown x_j from the ith equation. Note that this algorithm is applied to the augmented matrix, as the elements of the vector $\mathbf{b}$ are updated by the row operations as well.

It should be noted that in the above description of Gaussian elimination, each entry below the main diagonal is never explicitly zeroed, because that computation is unnecessary. It is only necessary to update entries of the matrix that are involved in subsequent row operations or the solution of the resulting upper triangular system. This system is solved by the following algorithm for *back substitution*. In the algorithm, we assume that U is the upper triangular matrix containing the coefficients of the system, and $\mathbf{y}$ is the vector containing the right-hand sides of the equations.

```
for i = n, n-1, ..., 1 do
    x_i = y_i
    for j = i+1, i+2, ..., n do
        x_i = x_i - u_ij x_j
    end
    x_i = x_i / u_ii
end
```

This algorithm requires approximately n^2 arithmetic operations. We will see that when solving systems of equations in which the right-hand side vector $\mathbf{b}$ is changing, but the coefficient matrix A remains fixed, it is quite practical to apply Gaussian elimination to A only once, and then repeatedly apply it to each $\mathbf{b}$, along with back substitution, because the latter two steps are much less expensive.

We now illustrate the use of both these algorithms with an example.

Example Consider the system of linear equations

$$\begin{aligned}
x_1 + 2x_2 + x_3 - x_4 &= 5 \\
3x_1 + 2x_2 + 4x_3 + 4x_4 &= 16 \\
4x_1 + 4x_2 + 3x_3 + 4x_4 &= 22 \\
2x_1 + x_3 + 5x_4 &= 15.
\end{aligned}$$

This system can be represented by the coefficient matrix A and right-hand side vector $\mathbf{b}$, as follows:

$$A = \begin{bmatrix} 1 & 2 & 1 & -1 \\ 3 & 2 & 4 & 4 \\ 4 & 4 & 3 & 4 \\ 2 & 0 & 1 & 5 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 5 \\ 16 \\ 22 \\ 15 \end{bmatrix}.$$

To perform row operations to reduce this system to upper triangular form, we define the augmented matrix

$$\tilde{A} = \left[\begin{array}{cc} A & \mathbf{b} \end{array} \right] = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 3 & 2 & 4 & 4 & 16 \\ 4 & 4 & 3 & 4 & 22 \\ 2 & 0 & 1 & 5 & 15 \end{bmatrix}.$$

We first define $\tilde{A}^{(1)} = \tilde{A}$ to be the original augmented matrix. Then, we denote by $\tilde{A}^{(2)}$ the result of the first elementary row operation, which entails subtracting 3 times the first row from the second in order to eliminate x_1 from the second equation:

$$\tilde{A}^{(2)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 4 & 4 & 3 & 4 & 22 \\ 2 & 0 & 1 & 5 & 15 \end{bmatrix}.$$

Next, we eliminate x_1 from the third equation by subtracting 4 times the first row from the third:

$$\tilde{A}^{(3)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 0 & -4 & -1 & 8 & 2 \\ 2 & 0 & 1 & 5 & 15 \end{bmatrix}.$$

Then, we complete the elimination of x_1 by subtracting 2 times the first row from the fourth:

$$\tilde{A}^{(4)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 0 & -4 & -1 & 8 & 2 \\ 0 & -4 & -1 & 7 & 5 \end{bmatrix}.$$

We now need to eliminate x_2 from the third and fourth equations. This is accomplished by subtracting the second row from the third, which yields

$$\tilde{A}^{(5)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 0 & 0 & -2 & 1 & 1 \\ 0 & -4 & -1 & 7 & 5 \end{bmatrix},$$

and the fourth, which yields

$$\tilde{A}^{(6)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 0 & 0 & -2 & 1 & 1 \\ 0 & 0 & -2 & 0 & 4 \end{bmatrix}.$$

Finally, we subtract the third row from the fourth to obtain the augmented matrix of an upper-triangular system,

$$\tilde{A}^{(7)} = \begin{bmatrix} 1 & 2 & 1 & -1 & 5 \\ 0 & -4 & 1 & 7 & 1 \\ 0 & 0 & -2 & 1 & 1 \\ 0 & 0 & 0 & -1 & 3 \end{bmatrix}.$$

Note that in a matrix for such a system, all entries below the *main diagonal* (the entries where the row index is equal to the column index) are equal to zero. That is, $a_{ij} = 0$ for $i > j$.

Now, we can perform back substitution on the corresponding system,

$$\begin{aligned} x_1 + 2x_2 + x_3 - x_4 &= 5, \\ -4x_2 + x_3 + 7x_4 &= 1, \\ -2x_3 + x_4 &= 1, \\ -x_4 &= 3, \end{aligned}$$

to obtain the solution, which yields $x_4 = -3$, $x_3 = -2$, $x_2 = -6$, and $x_1 = 16$. □

The *LU* Factorization

We have learned how to solve a system of linear equations $A\mathbf{x} = \mathbf{b}$ by applying Gaussian elimination to the augmented matrix $\tilde{A} = \begin{bmatrix} A & \mathbf{b} \end{bmatrix}$, and then performing back substitution on the resulting upper-triangular matrix. However, this approach is not practical if the right-hand side $\mathbf{b}$ of the system is changed, while A is not. This is due to the fact that the choice of $\mathbf{b}$ has no effect on the row operations needed to reduce A to upper-triangular form. Therefore, it is desirable to instead apply these row operations to A only once, and then "store" them in some way in order to apply them to any number of right-hand sides.

To accomplish this, we first note that subtracting m_{ij} times row j from row i to eliminate a_{ij}

is equivalent to multiplying A by the matrix

$$M_{ij} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & & & 0 \\ \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & -m_{ij} & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix},$$

where the entry $-m_{ij}$ is in row i, column j. More generally, if we let $A^{(1)} = A$ and let $A^{(k+1)}$ be the matrix obtained by eliminating elements of column k in $A^{(k)}$, then we have, for $k = 1, 2, \ldots, n-1$,

$$A^{(k+1)} = M^{(k)} A^{(k)}$$

where

$$M^{(k)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & & & & 0 \\ \vdots & \ddots & \ddots & \ddots & & & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & & & & \vdots \\ \vdots & & \vdots & -m_{k+1,k} & \ddots & \ddots & & & \vdots \\ \vdots & & \vdots & \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & 0 & -m_{nk} & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix},$$

with the elements $-m_{k+1,k}, \ldots, -m_{nk}$ occupying column k. It follows that the matrix

$$U = A^{(n)} = M^{(n-1)} A^{(n-1)} = M^{(n-1)} M^{(n-2)} \cdots M^{(1)} A$$

is upper triangular, and the vector

$$\mathbf{y} = M^{(n-1)} M^{(n-2)} \cdots M^{(1)} \mathbf{b},$$

being the result of applying the same row operations to $\mathbf{b}$, is the right-hand side for the upper-triangular system that is to be solved by back subtitution.

Unit Lower Triangular Matrices

We have previously learned about *upper triangular* matrices that result from Gaussian elimination. Recall that an $m \times n$ matrix A is upper triangular if $a_{ij} = 0$ whenever $i > j$. This means that all entries below the *main diagonal*, which consists of the entries a_{11}, a_{22}, $\ldots$, are equal to zero. A system of linear equations of the form $U\mathbf{x} = \mathbf{y}$, where U is an $n \times n$ nonsingular upper triangular matrix, can be solved by back substitution. Such a matrix is nonsingular if and only if all of its diagonal entries are nonzero.

Similarly, a matrix L is *lower triangular* if all of its entries above the main diagonal, that is, entries ℓ_{ij} for which $i < j$, are equal to zero. We will see that a system of equations of the form $L\mathbf{y} = \mathbf{b}$, where L is an $n \times n$ nonsingular lower triangular matrix, can be solved using a process similar to back substitution, called *forward substitution*. As with upper triangular matrices, a lower triangular matrix is nonsingular if and only if all of its diagonal entries are nonzero.

Triangular matrices have the following useful properties:

- The product of two upper (lower) triangular matrices is upper (lower) triangular.
- The inverse of a nonsingular upper (lower) triangular matrix is upper (lower) triangular.

That is, matrix multiplication and inversion preserve triangularity.

Now, we note that each matrix $M^{(k)}$, $k = 1, 2, \ldots, n-1$, is not only a lower-triangular matrix, but a *unit lower triangular* matrix, because all of its diagonal entries are equal to 1. Next, we note two important properties of unit lower (or upper) triangular matrices:

- The product of two unit lower (upper) triangular matrices is unit lower (upper) triangular.
- A unit lower (upper) triangular matrix is nonsingular, and its inverse is unit lower (upper) triangular.

In fact, the inverse of each $M^{(k)}$ is easily computed. We have

$$L^{(k)} = [M^{(k)}]^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & & & & 0 \\ \vdots & \ddots & \ddots & \ddots & & & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & & & & \vdots \\ \vdots & & \vdots & m_{k+1,k} & \ddots & \ddots & & & \vdots \\ \vdots & & \vdots & \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & 0 & m_{nk} & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}.$$

It follows that if we define $M = M^{(n-1)} \cdots M^{(1)}$, then M is unit lower triangular, and $MA = U$, where U is upper triangular. It follows that $A = M^{-1}U = LU$, where

$$L = L^{(1)} \cdots L^{(n-1)} = [M^{(1)}]^{-1} \cdots [M^{(n-1)}]^{-1}$$

is also unit lower triangular. Furthermore, from the structure of each matrix $L^{(k)}$, it can readily be determined that

$$L = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ m_{21} & 1 & 0 & & \vdots \\ \vdots & m_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{bmatrix}.$$

That is, L stores all of the multipliers used during Gaussian elimination. The factorization of A that we have obtained,

$$A = LU,$$

is called the *LU decomposition*, or *LU factorization*, of A.

Solution of $A\mathbf{x} = \mathbf{b}$

Once the LU decomposition $A = LU$ has been computed, we can solve the system $A\mathbf{x} = \mathbf{b}$ by first noting that if $\mathbf{x}$ is the solution, then

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b}.$$

Therefore, we can obtain $\mathbf{x}$ by first solving the system

$$L\mathbf{y} = \mathbf{b},$$

and then solving

$$U\mathbf{x} = \mathbf{y}.$$

Then, if $\mathbf{b}$ should change, then only these last two systems need to be solved in order to obtain the new solution; the LU decomposition does not need to be recomputed.

The system $U\mathbf{x} = \mathbf{y}$ can be solved by back substitution, since U is upper-triangular. To solve $L\mathbf{y} = \mathbf{b}$, we can use *forward substitution*, since L is unit lower triangular.

for $i = 1, 2, \ldots, n$ **do**
  $y_i = b_i$
  for $j = 1, 2, \ldots, i-1$ **do**
    $y_i = y_i - \ell_{ij} y_j$
  end
end

Like back substitution, this algorithm requires $O(n^2)$ floating-point operations. Unlike back substitution, there is no division of the ith component of the solution by a diagonal element of the matrix, but this is only because in this context, L is unit lower triangular, so $\ell_{ii} = 1$. When applying forward substitution to a general lower triangular matrix, such a division is required.

Implementation Details

Because both forward and back substituion require only $O(n^2)$ operations, whereas Gaussian elimination requires $O(n^3)$ operations, changes in the right-hand side $\mathbf{b}$ can be handled quite efficiently by computing the factors L and U once, and storing them. This can be accomplished quite efficiently, because L is unit lower triangular. It follows from this that L and U can be stored in a single $n \times n$ matrix by storing U in the upper triangular part, and the multipliers m_{ij} in the lower triangular part.

Existence and Uniqueness

Not every nonsingular $n \times n$ matrix A has an LU decomposition. For example, if $a_{11} = 0$, then the multipliers m_{i1}, for $i = 2, 3, \ldots, n$, are not defined, so no multiple of the first row can be added to the other rows to eliminate subdiagonal elements in the first column. That is, Gaussian elimination can break down. Even if $a_{11} \neq 0$, it can happen that the (j, j) element of $A^{(j)}$ is zero, in which case a similar breakdown occurs. When this is the case, the LU decomposition of A does not exist. This will be addressed by *pivoting*, resulting in a modification of the LU decomposition.

It can be shown that the LU decomposition of an $n \times n$ matrix A *does* exist if the *leading principal minors* of A, defined by

$$[A]_{1:k,1:k} = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix}, \quad k = 1, 2, \ldots, n,$$

are all nonsingular. Furthermore, when the LU decomposition exists, it is unique. To see this, suppose that A has two distinct LU decompositions, $A = L_1U_1 = L_2U_2$. Then, because the factors of each decomposition are nonsingular, it follows that

$$L_2^{-1}L_1 = U_2U_1^{-1}.$$

The left side is lower-triangular, while right side is upper-triangular, because the triangularity of a matrix is preserved by inversion and multiplication with matrices of the same triangularity.

It follows that both matrices must be diagonal. However, the left side is also *unit* lower triangular, and so both sides must equal the identity matrix I. Therefore, $L_2 = L_1$ and $U_1 = U_2$, contradicting the assumption that the decompositions are distinct. We conclude that the LU decomposition is unique.

Practical Computation of Determinants

Computing the determinant of an $n \times n$ matrix A using its definition requires a number of arithmetic operations that is exponential in n. However, more practical methods for computing the determinant can be obtained by using its properties:

- If $\tilde{A}$ is obtained from A by adding a multiple of a row of A to another row, then $\det(\tilde{A}) = \det(A)$.
- If B is an $n \times n$ matrix, then $\det(AB) = \det(A)\det(B)$.
- If A is a triangular matrix (either upper or lower), then $\det(A) = \prod_{i=1}^{n} a_{ii}$.

It follows from these properties that if Gaussian elimination is used to reduce A to an upper-triangular matrix U, then $\det(A) = \det(U)$, where U is the resulting upper-triangular matrix, because the elementary row operations needed to reduce A to U do not change the determinant. Because U is upper triangular, $\det(U)$, being the product of its diagonal entries, can be computed in $n - 1$ multiplications. It follows that the determinant of any matrix can be computed in $O(n^3)$ operations.

It can also be seen that $\det(A) = \det(U)$ by noting that if $A = LU$, then $\det(A) = \det(L)\det(U)$, by one of the abovementioned properties, but $\det(L) = 1$, because L is a unit lower triangular matrix. It follows from the fact that L is lower triangular that $\det(L)$ is the product of its diagonal entries, and it follows from the fact that L is *unit* lower triangular that all of its diagonal entries are equal to 1.